

Uma introdução a Metaprogramação (com Python)

Quem sou eu

- Geraldo que não é Geraldo
- Desenvolvedor
- Grupy-RN & Grupy-Floripa
- Mossoró/RN - Floripa/SC
- Diabético & Vegetariano
- Curioso e hiperativo



Conteúdo

- De onde surgiu a ideia?
- Considerações iniciais
- O que é?
 - O que não é
 - Definições
 - Casos
 - Vantagens e desvantagens
- Onde vivem?
 - #define (C)
 - eval (Python)
 - decorators (Python)
 - metaclasses (Python)
 - macros (Elixir)
- Do que se alimentam?
- Como se reproduzem?
- Python e metaprogramação
 - eval
 - exec
 - decorators
 - interceptação
 - descritores
 - metaclasses
- Material extra
- Referencias

Material



bit.ly/exageraldo-na-scc-uern-22

De onde surgiu a ideia?

- Sou um tanto curioso
- Amo estudar linguagens de programação novas
- Frasko & NoSeiQueLa ORM 🤪
- "por que não estudar pra apresentar?!"



Toddy parabens



Considerações iniciais

- Duração estimada: 50/60 min
- Não é um material aprofundado, mas sim um roteiro para guiar uma conversa
- É a primeira vez que eu converso/apresento sobre esse assunto
- Muito provavelmente vou devagar/brisar entre alguns assuntos.
- Para melhor compreender esse conteúdo, é esperado/desejado:
 - Conhecimento/experiencia na linguagem de programação Python
 - Alguns conceitos de computação (POO/funcional, macros, compilar/interpretar, ...)

O que é?

O que não é

The image shows a blue square containing the word "Meta" in a white, sans-serif font. This is the standard Meta logo.

Definições

Meta: Prefixo usado para significar sobre (sua própria categoria)

- metadados -> dados sobre dados
- metadiscussão -> discussão sobre discussão
- metapiada -> piada sobre piadas
- **metaprogramação -> escrever programas que escrevem programas**

Definições

- Metaprogramação é a programação de programas que escrevem ou manipulam outros programas (ou a si próprios) assim como seus dados, ou que fazem parte do trabalho em tempo de compilação.
- De um modo geral, metaprogramação é toda programação que atua sobre outro programa, seja em formato fonte, binário, ou numa representação abstrata em memória.
- Técnica de especificar modelos genéricos de código fonte a partir dos quais classes de componentes, ou partes deles, podem ser instanciadas automaticamente para produzir novos componentes de software.
- Metaprogramação é escrever programas onde eles mesmos escrevem código.

Casos

- geração de compiladores
- geração de aplicações
- análise de código
- criação de componentes genéricos
- transformações/modificações de programas
- manutenção/evolução/configuração de software
- otimização preventiva
- padrões de projeto
- avaliação parcial

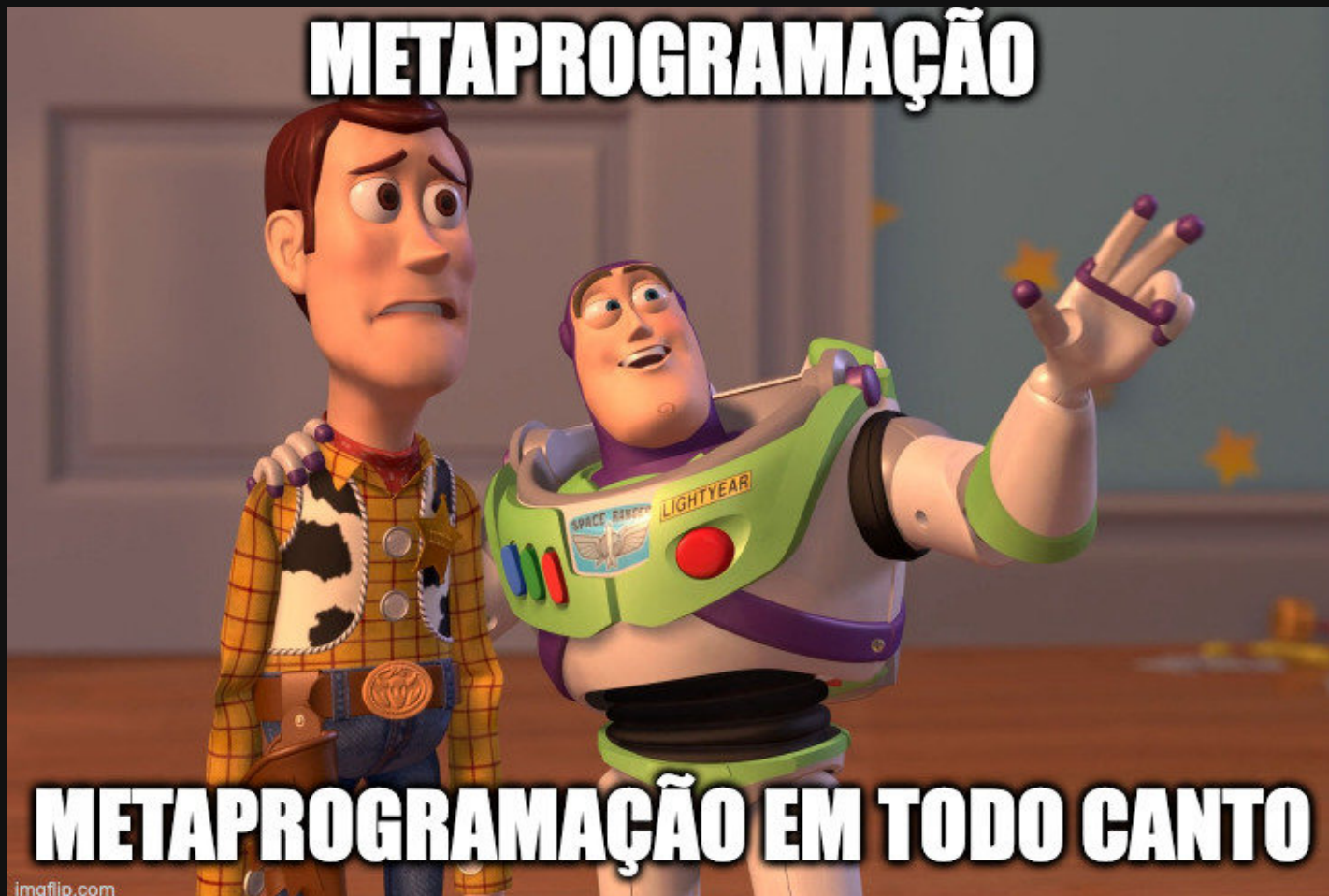
Vantagens

- Extensível
- Precisão
- Fundamentação
- Menos código
- Performance

Desvantagens

- Dificuldade
- Segurança
- Exceções
- Erros invisíveis
- Strings

Onde vivem?



Compiladores, transpiladores e interpretadores são bons exemplos de metaprogramas. Eles tomam programas em uma forma e os transformam em código de máquina, bytecode ou mesmo código fonte em outra língua.

#define (c)

```
#include <stdio.h>
```

```
#define ADD(a, b) ((a) + (b))
```

```
int main() {
```

```
    int value = ADD(10, 20);
```

```
    printf("A soma deu %i", value);
```

```
    return 0;
```

```
}
```

```
// A soma deu 30
```


eval (python)

```
eval("10+10")
```

```
# 20
```

decorators (python)

```
func_um = func_decorator(func_um)
```

metaclasses (python)

```
class SingletonMeta(type):  
    _instances = {}  
    def __call__(cls, *args, **kwargs):  
        if cls not in cls._instances:  
            cls._instances[cls] = super(  
                SingletonMeta,  
                cls  
            ).__call__(*args, **kwargs)  
        return cls._instances[cls]
```

```
class SingletonClass(metaclass=SingletonMeta):  
    pass
```

macros (elixir)

```
defmodule Foo do
  defmacro macro_inspect(value) do
    IO.inspect(value)
    value
  end

  def func_inspect(value) do
    IO.inspect(value)
    value
  end
end
```

Do que se alimentam?

da sua mente

corpo

e alma



Como se reproduzem?



Python e metaprogramação

Mão na massa (ou quase)



Vamos para o Jupyter

MUITO OBRIGADO



PERGUNTAS?